In [2]:

```
import numpy as np
import pandas as pd
import warnings

warnings.filterwarnings('ignore')

# Generating synthetic data
np.random.seed(0)
data_size = 200
features = np.random.rand(data_size, 2)  # Two features: visit duration and
labels = (features[:, 0] + features[:, 1] > 1).astype(int)  # Purchase (1) c

# Convert to DataFrame for easier manipulation
df = pd.DataFrame(features, columns=['VisitDuration', 'PagesVisited'])
df['Purchase'] = labels

# Print the 'features' array
print("--- Features Array (NumPy) ---")
print(features)
print("\nShape of features array:", features.shape)
print("Type of features array:", type(features))

# Print the 'df' DataFrame
print("\n--- DataFrame (Pandas) ---")
print(df)
print("\nShape of DataFrame:", df.shape)
print("Type of DataFrame:", type(df))
print("\nInfo of DataFrame:")
df.info()
print("\nHead of DataFrame:")
print(df.head())
print("\nTail of DataFrame:")
print(df.tail())
```

```
--- Features Array (NumPy) ---
[[0.5488135  0.71518937]
 [0.60276338 0.54488318]
 [0.4236548  0.64589411]
 [0.43758721 0.891773  ]
 [0.96366276 0.38344152]
 [0.79172504 0.52889492]
 [0.56804456 0.92559664]
 [0.07103606 0.0871293 ]
 [0.0202184  0.83261985]
 [0.77815675 0.87001215]
 [0.97861834 0.79915856]
 [0.46147936 0.78052918]
 [0.11827443 0.63992102]
 [0.14335329 0.94466892]
 [0.52184832 0.41466194]
 [0.26455561 0.77423369]
 [0.45615033 0.56843395]
 [0.0187898  0.6176355 ]
 [0.61209572 0.616934  ]
 [0.94374808 0.6818203 ]
 [0.3595079  0.43703195]
 [0.6976312  0.06022547]
 [0.66676672 0.67063787]
 [0.21038256 0.1289263 ]
 [0.31542835 0.36371077]
 [0.57019677 0.43860151]
 [0.98837384 0.10204481]
 [0.20887676 0.16130952]
 [0.65310833 0.2532916 ]
 [0.46631077 0.24442559]
 [0.15896958 0.11037514]
 [0.65632959 0.13818295]
 [0.19658236 0.36872517]
 [0.82099323 0.09710128]
 [0.83794491 0.09609841]
 [0.97645947 0.4686512 ]
 [0.97676109 0.60484552]
 [0.73926358 0.03918779]
 [0.28280696 0.12019656]
 [0.2961402  0.11872772]
 [0.31798318 0.41426299]
 [0.0641475  0.69247212]
 [0.56660145 0.26538949]
 [0.52324805 0.09394051]
 [0.5759465  0.9292962 ]
 [0.31856895 0.66741038]
 [0.13179786 0.7163272 ]
 [0.28940609 0.18319136]
 [0.58651293 0.02010755]
 [0.82894003 0.00469548]
 [0.67781654 0.27000797]
 [0.73519402 0.96218855]
 [0.24875314 0.57615733]
 [0.59204193 0.57225191]
 [0.22308163 0.95274901]
```

```
[0.44712538 0.84640867]
[0.69947928 0.29743695]
[0.81379782 0.39650574]
[0.8811032  0.58127287]
[0.88173536 0.69253159]
[0.72525428 0.50132438]
[0.95608363 0.6439902 ]
[0.42385505 0.60639321]
[0.0191932  0.30157482]
[0.66017354 0.29007761]
[0.61801543 0.4287687 ]
[0.13547406 0.29828233]
[0.56996491 0.59087276]
[0.57432525 0.65320082]
[0.65210327 0.43141844]
[0.8965466  0.36756187]
[0.43586493 0.89192336]
[0.80619399 0.70388858]
[0.10022689 0.91948261]
[0.7142413  0.99884701]
[0.1494483  0.86812606]
[0.16249293 0.61555956]
[0.12381998 0.84800823]
[0.80731896 0.56910074]
[0.4071833  0.069167  ]
[0.69742877 0.45354268]
[0.7220556  0.86638233]
[0.97552151 0.85580334]
[0.01171408 0.35997806]
[0.72999056 0.17162968]
[0.52103661 0.05433799]
[0.19999652 0.01852179]
[0.7936977  0.22392469]
[0.34535168 0.92808129]
[0.7044144  0.03183893]
[0.16469416 0.6214784 ]
[0.57722859 0.23789282]
[0.934214   0.61396596]
[0.5356328  0.58990998]
[0.73012203 0.311945  ]
[0.39822106 0.20984375]
[0.18619301 0.94437239]
[0.7395508  0.49045881]
[0.22741463 0.25435648]
[0.05802916 0.43441663]
[0.31179588 0.69634349]
[0.37775184 0.17960368]
[0.02467873 0.06724963]
[0.67939277 0.45369684]
[0.53657921 0.89667129]
[0.99033895 0.21689698]
[0.6630782  0.26332238]
[0.020651   0.75837865]
[0.32001715 0.38346389]
[0.58831711 0.83104846]
[0.62898184 0.87265066]
```

```
[0.27354203 0.79804683]
[0.18563594 0.95279166]
[0.68748828 0.21550768]
[0.94737059 0.73085581]
[0.25394164 0.21331198]
[0.51820071 0.02566272]
[0.20747008 0.42468547]
[0.37416998 0.46357542]
[0.27762871 0.58678435]
[0.86385561 0.11753186]
[0.51737911 0.13206811]
[0.71685968 0.3960597 ]
[0.56542131 0.18327984]
[0.14484776 0.48805628]
[0.35561274 0.94043195]
[0.76532525 0.74866362]
[0.90371974 0.08342244]
[0.55219247 0.58447607]
[0.96193638 0.29214753]
[0.24082878 0.10029394]
[0.01642963 0.92952932]
[0.66991655 0.78515291]
[0.28173011 0.58641017]
[0.06395527 0.4856276 ]
[0.97749514 0.87650525]
[0.33815895 0.96157015]
[0.23170163 0.94931882]
[0.9413777  0.79920259]
[0.63044794 0.87428797]
[0.29302028 0.84894356]
[0.61787669 0.01323686]
[0.34723352 0.14814086]
[0.98182939 0.47837031]
[0.49739137 0.63947252]
[0.36858461 0.13690027]
[0.82211773 0.18984791]
[0.51131898 0.22431703]
[0.09784448 0.86219152]
[0.97291949 0.96083466]
[0.9065555  0.77404733]
[0.33314515 0.08110139]
[0.40724117 0.23223414]
[0.13248763 0.05342718]
[0.72559436 0.01142746]
[0.77058075 0.14694665]
[0.07952208 0.08960303]
[0.67204781 0.24536721]
[0.42053947 0.55736879]
[0.86055117 0.72704426]
[0.27032791 0.1314828 ]
[0.05537432 0.30159863]
[0.26211815 0.45614057]
[0.68328134 0.69562545]
[0.28351885 0.37992696]
[0.18115096 0.78854551]
[0.05684808 0.69699724]
```

```
 [0.7786954  0.77740756]
 [0.25942256 0.37381314]
 [0.58759964 0.2728219 ]
 [0.3708528  0.19705428]
 [0.45985588 0.0446123 ]
 [0.79979588 0.07695645]
 [0.51883515 0.3068101 ]
 [0.57754295 0.95943334]
 [0.64557024 0.03536244]
 [0.43040244 0.51001685]
 [0.53617749 0.68139251]
 [0.2775961  0.12886057]
 [0.39267568 0.95640572]
 [0.18713089 0.90398395]
 [0.54380595 0.45691142]
 [0.88204141 0.45860396]
 [0.72416764 0.39902532]
 [0.90404439 0.69002502]
 [0.69962205 0.3277204 ]
 [0.75677864 0.63606106]
 [0.24002027 0.16053882]
 [0.79639147 0.9591666 ]
 [0.45813883 0.59098417]
 [0.85772264 0.45722345]
 [0.95187448 0.57575116]
 [0.82076712 0.90884372]
 [0.81552382 0.15941446]
 [0.62889844 0.39843426]
 [0.06271295 0.42403225]
 [0.25868407 0.84903831]
 [0.03330463 0.95898272]
 [0.35536885 0.35670689]
 [0.0163285  0.18523233]]

Shape of features array: (200, 2)
Type of features array: <class 'numpy.ndarray'>

--- DataFrame (Pandas) ---
     VisitDuration  PagesVisited  Purchase
0         0.548814      0.715189         1
1         0.602763      0.544883         1
2         0.423655      0.645894         1
3         0.437587      0.891773         1
4         0.963663      0.383442         1
..             ...           ...       ...
195       0.062713      0.424032         0
196       0.258684      0.849038         1
197       0.033305      0.958983         0
198       0.355369      0.356707         0
199       0.016329      0.185232         0

[200 rows x 3 columns]

Shape of DataFrame: (200, 3)
Type of DataFrame: <class 'pandas.core.frame.DataFrame'>
```

```
Info of DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   VisitDuration  200 non-null    float64
 1   PagesVisited   200 non-null    float64
 2   Purchase       200 non-null    int64
dtypes: float64(2), int64(1)
memory usage: 4.8 KB

Head of DataFrame:
   VisitDuration  PagesVisited  Purchase
0       0.548814      0.715189         1
1       0.602763      0.544883         1
2       0.423655      0.645894         1
3       0.437587      0.891773         1
4       0.963663      0.383442         1

Tail of DataFrame:
     VisitDuration  PagesVisited  Purchase
195       0.062713      0.424032         0
196       0.258684      0.849038         1
197       0.033305      0.958983         0
198       0.355369      0.356707         0
199       0.016329      0.185232         0
```

In [4]:

```
from sklearn.model_selection import train_test_split

# Split the data
X_train, X_test, y_train, y_test = train_test_split(df[['VisitDuration', 'Pa
```

In [16]:

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Define the model
model = Sequential([
    Dense(10, activation='relu', input_shape=(2,)),  # Input layer with 2 fe
    Dense(1, activation='sigmoid')  # Output layer with sigmoid activation f
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accura

#Supress Warnings
import warnings
warnings.filterwarnings('ignore')

# Train the model and save the history
history = model.fit(X_train, y_train, epochs=100, batch_size=5, validation_s

# Plotting training and validation loss and accuracy
import matplotlib.pyplot as plt
```

```python
plt.figure(figsize=(12, 5))

# Plot training & validation accuracy values
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')

# Plot training & validation loss values
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')

plt.show()
```

```
Epoch 1/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.4228 - loss: 0.7097 - v
al_accuracy: 0.4688 - val_loss: 0.7068
Epoch 2/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5021 - loss: 0.6910 - va
l_accuracy: 0.4688 - val_loss: 0.7001
Epoch 3/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5184 - loss: 0.6904 - va
l_accuracy: 0.5000 - val_loss: 0.6932
Epoch 4/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.5974 - loss: 0.6846 - va
l_accuracy: 0.5312 - val_loss: 0.6872
Epoch 5/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6526 - loss: 0.6761 - va
l_accuracy: 0.5938 - val_loss: 0.6806
Epoch 6/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6995 - loss: 0.6601 - va
l_accuracy: 0.6250 - val_loss: 0.6752
Epoch 7/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6589 - loss: 0.6675 - va
l_accuracy: 0.6250 - val_loss: 0.6691
Epoch 8/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7045 - loss: 0.6606 - va
l_accuracy: 0.6875 - val_loss: 0.6633
Epoch 9/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6458 - loss: 0.6708 - va
l_accuracy: 0.6875 - val_loss: 0.6567
Epoch 10/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7142 - loss: 0.6530 - va
l_accuracy: 0.7188 - val_loss: 0.6515
Epoch 11/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7331 - loss: 0.6531 - va
l_accuracy: 0.6875 - val_loss: 0.6448
Epoch 12/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7716 - loss: 0.6419 - va
l_accuracy: 0.6875 - val_loss: 0.6390
Epoch 13/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7864 - loss: 0.6410 - va
l_accuracy: 0.7188 - val_loss: 0.6332
Epoch 14/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.7321 - loss: 0.6506 - va
l_accuracy: 0.7188 - val_loss: 0.6276
Epoch 15/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8160 - loss: 0.6244 - va
l_accuracy: 0.6875 - val_loss: 0.6205
Epoch 16/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7979 - loss: 0.6307 - va
l_accuracy: 0.7500 - val_loss: 0.6147
Epoch 17/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8115 - loss: 0.6137 - va
l_accuracy: 0.7188 - val_loss: 0.6077
Epoch 18/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8515 - loss: 0.6100 - va
l_accuracy: 0.7500 - val_loss: 0.6016
Epoch 19/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8644 - loss: 0.5901 - va
```

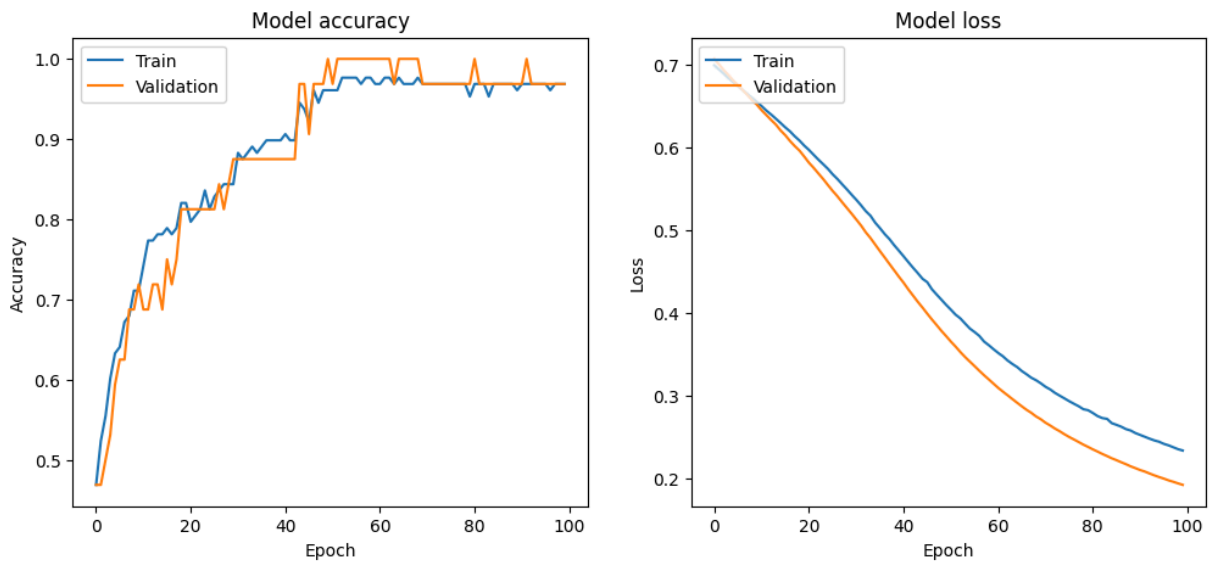
```
l_accuracy: 0.8125 - val_loss: 0.5961
Epoch 20/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8069 - loss: 0.6049 - va
l_accuracy: 0.8125 - val_loss: 0.5890
Epoch 21/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8287 - loss: 0.5939 - va
l_accuracy: 0.8125 - val_loss: 0.5815
Epoch 22/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7835 - loss: 0.6001 - va
l_accuracy: 0.8125 - val_loss: 0.5753
Epoch 23/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8466 - loss: 0.5873 - va
l_accuracy: 0.8125 - val_loss: 0.5685
Epoch 24/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8285 - loss: 0.5840 - va
l_accuracy: 0.8125 - val_loss: 0.5619
Epoch 25/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8488 - loss: 0.5550 - va
l_accuracy: 0.8125 - val_loss: 0.5546
Epoch 26/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8318 - loss: 0.5608 - va
l_accuracy: 0.8125 - val_loss: 0.5476
Epoch 27/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.7919 - loss: 0.5701 - va
l_accuracy: 0.8438 - val_loss: 0.5411
Epoch 28/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8480 - loss: 0.5554 - va
l_accuracy: 0.8125 - val_loss: 0.5340
Epoch 29/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8715 - loss: 0.5460 - va
l_accuracy: 0.8438 - val_loss: 0.5271
Epoch 30/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8458 - loss: 0.5348 - va
l_accuracy: 0.8750 - val_loss: 0.5202
Epoch 31/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8598 - loss: 0.5479 - va
l_accuracy: 0.8750 - val_loss: 0.5130
Epoch 32/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8921 - loss: 0.5433 - va
l_accuracy: 0.8750 - val_loss: 0.5059
Epoch 33/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8881 - loss: 0.5259 - va
l_accuracy: 0.8750 - val_loss: 0.4976
Epoch 34/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8511 - loss: 0.5365 - va
l_accuracy: 0.8750 - val_loss: 0.4907
Epoch 35/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8725 - loss: 0.5286 - va
l_accuracy: 0.8750 - val_loss: 0.4828
Epoch 36/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8795 - loss: 0.5085 - va
l_accuracy: 0.8750 - val_loss: 0.4749
Epoch 37/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9206 - loss: 0.4931 - va
l_accuracy: 0.8750 - val_loss: 0.4673
Epoch 38/100
```

```
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9340 - loss: 0.4986 - va
l_accuracy: 0.8750 - val_loss: 0.4595
Epoch 39/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9238 - loss: 0.4738 - va
l_accuracy: 0.8750 - val_loss: 0.4518
Epoch 40/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9011 - loss: 0.4752 - va
l_accuracy: 0.8750 - val_loss: 0.4441
Epoch 41/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8759 - loss: 0.4686 - va
l_accuracy: 0.8750 - val_loss: 0.4368
Epoch 42/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9245 - loss: 0.4668 - va
l_accuracy: 0.8750 - val_loss: 0.4287
Epoch 43/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9240 - loss: 0.4471 - va
l_accuracy: 0.8750 - val_loss: 0.4212
Epoch 44/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9179 - loss: 0.4698 - va
l_accuracy: 0.9688 - val_loss: 0.4137
Epoch 45/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9128 - loss: 0.4496 - va
l_accuracy: 0.9688 - val_loss: 0.4068
Epoch 46/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9153 - loss: 0.4539 - va
l_accuracy: 0.9062 - val_loss: 0.3994
Epoch 47/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9759 - loss: 0.4187 - va
l_accuracy: 0.9688 - val_loss: 0.3923
Epoch 48/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9526 - loss: 0.4264 - va
l_accuracy: 0.9688 - val_loss: 0.3856
Epoch 49/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9646 - loss: 0.4156 - va
l_accuracy: 0.9688 - val_loss: 0.3787
Epoch 50/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9556 - loss: 0.4065 - va
l_accuracy: 1.0000 - val_loss: 0.3723
Epoch 51/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9612 - loss: 0.4084 - va
l_accuracy: 0.9688 - val_loss: 0.3657
Epoch 52/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9666 - loss: 0.4058 - va
l_accuracy: 1.0000 - val_loss: 0.3597
Epoch 53/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9701 - loss: 0.3912 - va
l_accuracy: 1.0000 - val_loss: 0.3531
Epoch 54/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9824 - loss: 0.3910 - va
l_accuracy: 1.0000 - val_loss: 0.3471
Epoch 55/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9652 - loss: 0.3872 - va
l_accuracy: 1.0000 - val_loss: 0.3416
Epoch 56/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9427 - loss: 0.3824 - va
l_accuracy: 1.0000 - val_loss: 0.3361
```

```
Epoch 57/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9789 - loss: 0.3604 - va
l_accuracy: 1.0000 - val_loss: 0.3306
Epoch 58/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9917 - loss: 0.3550 - va
l_accuracy: 1.0000 - val_loss: 0.3252
Epoch 59/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9417 - loss: 0.3670 - va
l_accuracy: 1.0000 - val_loss: 0.3202
Epoch 60/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9478 - loss: 0.3689 - va
l_accuracy: 1.0000 - val_loss: 0.3150
Epoch 61/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9710 - loss: 0.3648 - va
l_accuracy: 1.0000 - val_loss: 0.3099
Epoch 62/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9699 - loss: 0.3520 - va
l_accuracy: 1.0000 - val_loss: 0.3053
Epoch 63/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9691 - loss: 0.3717 - va
l_accuracy: 1.0000 - val_loss: 0.3009
Epoch 64/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9799 - loss: 0.3383 - va
l_accuracy: 0.9688 - val_loss: 0.2964
Epoch 65/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9799 - loss: 0.3429 - va
l_accuracy: 1.0000 - val_loss: 0.2920
Epoch 66/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9628 - loss: 0.3715 - va
l_accuracy: 1.0000 - val_loss: 0.2876
Epoch 67/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9510 - loss: 0.3186 - va
l_accuracy: 1.0000 - val_loss: 0.2834
Epoch 68/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9535 - loss: 0.3646 - va
l_accuracy: 1.0000 - val_loss: 0.2797
Epoch 69/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9925 - loss: 0.3054 - va
l_accuracy: 1.0000 - val_loss: 0.2754
Epoch 70/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9379 - loss: 0.3195 - va
l_accuracy: 0.9688 - val_loss: 0.2721
Epoch 71/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9810 - loss: 0.2985 - va
l_accuracy: 0.9688 - val_loss: 0.2678
Epoch 72/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9636 - loss: 0.3114 - va
l_accuracy: 0.9688 - val_loss: 0.2645
Epoch 73/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9661 - loss: 0.2895 - va
l_accuracy: 0.9688 - val_loss: 0.2608
Epoch 74/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9757 - loss: 0.3078 - va
l_accuracy: 0.9688 - val_loss: 0.2576
Epoch 75/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9929 - loss: 0.2695 - va
```

```
l_accuracy: 0.9688 - val_loss: 0.2539
Epoch 76/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9858 - loss: 0.2837 - va
l_accuracy: 0.9688 - val_loss: 0.2506
Epoch 77/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9651 - loss: 0.2917 - va
l_accuracy: 0.9688 - val_loss: 0.2476
Epoch 78/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9645 - loss: 0.2860 - va
l_accuracy: 0.9688 - val_loss: 0.2446
Epoch 79/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9597 - loss: 0.2807 - va
l_accuracy: 0.9688 - val_loss: 0.2414
Epoch 80/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9485 - loss: 0.2869 - va
l_accuracy: 0.9688 - val_loss: 0.2386
Epoch 81/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9819 - loss: 0.2797 - va
l_accuracy: 1.0000 - val_loss: 0.2357
Epoch 82/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9723 - loss: 0.2734 - va
l_accuracy: 0.9688 - val_loss: 0.2330
Epoch 83/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9524 - loss: 0.2744 - va
l_accuracy: 0.9688 - val_loss: 0.2302
Epoch 84/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9555 - loss: 0.2639 - va
l_accuracy: 0.9688 - val_loss: 0.2276
Epoch 85/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9648 - loss: 0.2763 - va
l_accuracy: 0.9688 - val_loss: 0.2249
Epoch 86/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9601 - loss: 0.2556 - va
l_accuracy: 0.9688 - val_loss: 0.2227
Epoch 87/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9561 - loss: 0.2864 - va
l_accuracy: 0.9688 - val_loss: 0.2201
Epoch 88/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9676 - loss: 0.2718 - va
l_accuracy: 0.9688 - val_loss: 0.2178
Epoch 89/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9853 - loss: 0.2702 - va
l_accuracy: 0.9688 - val_loss: 0.2152
Epoch 90/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9684 - loss: 0.2467 - va
l_accuracy: 0.9688 - val_loss: 0.2130
Epoch 91/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9778 - loss: 0.2464 - va
l_accuracy: 0.9688 - val_loss: 0.2107
Epoch 92/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9413 - loss: 0.2864 - va
l_accuracy: 1.0000 - val_loss: 0.2089
Epoch 93/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9560 - loss: 0.2534 - va
l_accuracy: 0.9688 - val_loss: 0.2066
Epoch 94/100
```

```
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9899 - loss: 0.2281 - va
l_accuracy: 0.9688 - val_loss: 0.2043
Epoch 95/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9790 - loss: 0.2548 - va
l_accuracy: 0.9688 - val_loss: 0.2023
Epoch 96/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9555 - loss: 0.2629 - va
l_accuracy: 0.9688 - val_loss: 0.2004
Epoch 97/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9683 - loss: 0.2434 - va
l_accuracy: 0.9688 - val_loss: 0.1983
Epoch 98/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9598 - loss: 0.2199 - va
l_accuracy: 0.9688 - val_loss: 0.1966
Epoch 99/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9669 - loss: 0.2233 - va
l_accuracy: 0.9688 - val_loss: 0.1947
Epoch 100/100
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9765 - loss: 0.2298 - va
l_accuracy: 0.9688 - val_loss: 0.1929
```



In [17]:
```python
# Evaluate the model on the test set
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {accuracy}")
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step - accuracy: 0.9187 - loss: 0.2645
Test Accuracy: 0.925000011920929
```